

DATORÖVNING 3 — BISEKTIONSALGORITMEN

Allmänt. Dokumentera ditt arbete i ett pdf-dokument. Spara alla filer så att du kan läsa på inför tentamen. Det räcker inte att din kompis har filerna. Datorövningarna examineras vid tentamen. Samarbete uppmuntras, men detta är inget grupparbete. Varje student måste göra sina egna datorprogram och sina egna dokument.

Mål.

Matematik: bisektionsalgoritmen, konvergent följd.

Programmering: `while`-loop, `if`, funktion som argument till funktion.

Litteratur. BM kap 4, Jönsson, kap. 6,7.

Instruktioner. Skapa en ny filkatalog `studio-3` för denna övning.

Dokumentera allt ditt arbete i ett word-dokument. Exportera slutligen word-dokumentet till pdf-format. Det görs med `'print'` och skrivare `'PDF995'`. Slutprodukten skall alltså vara pdf. Visa upp för din lärare som prickar av att du gjort övningen.

Demonstration. Läraren börjar med att demonstrera följande program:

```
function y = teckentest(f, a, b)
% teckentest - kollar om f(a) och f(b) har samma tecken
%
% Syntax:
%       y = teckentest(f, a, b)
% Input:
%       f - funktionshandtag
%       a, b - två tal
% Output:
%       y - ett tal
% Description:
%       Programmet returnerar värdet 1 om f(a) och f(b) har samma tecken.
%       Programmet returnerar värdet -1 om f(a) och f(b) har olika tecken.
%       Programmet returnerar värdet 0 om en av f(a) och f(b) är noll.
%
%-----
```

```
y=0;
if f(a)*f(b)<0
    y=-1;
end
if f(a)*f(b)>0
    y=1;
end
```

Kommandorad:

```
>> y = teckentest(@sin, 1, 2)
>> y = teckentest(@funk1, 1, 2)
```

Här måste man först skriva funktionsfilen `funk1.m`.

Uppgifter.

1. Skriv ett program som implementerar bisektionsalgoritmen enligt programskalet [bisect.m](#) och flödesschemat [bisect.pdf](#). Kopiera filen genom att klicka på denna länk eller gå till länken “matlab” på kurshemsidan och ladda ned den därifrån.

2. Använd programmet för att beräkna alla rötter till följande ekvationer. Tips: skriv ekvationerna på formen $f(x) = 0$, skriv en funktionsfil för varje exempel, kalla dem `funk1.m`, `funk2.m` och så vidare, plotta funktionerna för att få en grov uppfattning om var rötterna ligger.

- (1) $x^2 = 2$
- (2) $x^4 = 2$
- (3) $x^2 - 4x + 8 = 0$
- (4) $\sin(x) = 0$
- (5) $\cos(x) = x$
- (6) $11x^3 - 2x^2 - 77x + 14 = 0$
- (7) Hitta på en ekvation själv.

Tips: när funktionen finns som en funktionsfil är det bekvämt att använda `fplot`:

```
>> fplot(@funk1, [-5, 7])
```

3. Lägg till kod som skriver ut ett meddelande på skärmen och sedan stoppar och returnerar ett tomt värde `x = []` om $f(a)$ och $f(b)$ inte har olika tecken.

4. Skriv ett program som implementerar bisektionsalgoritmen enligt programskalet [bisectdemo.m](#) och [detta flödesschema](#) (“matlab/bisectdemo.pdf”). Obs att skillnaden jämfört med `bisect.m` är att vi nu sparar följderna x_i och X_i i varsin kolonnvektor. (Anledningen till att vi skriver två nästan likadana program är att du ska förstå programmeringen och matematiken bättre.)

5. Kör programmet på något av föregående exempel. Observera hur de beräknade följderna konvergerar. Hur många steg behöver algoritmen för att felet ska bli mindre än 10^{-10} ?

Jämför med det teoretiska resultatet $|x_i - X_i| \leq (b - a)2^{-i} \leq 10^{-10}$. Tips: lös ut antalet steg i ur denna olikhet.

6. MATLAB har programmet `fsolve`. Läs om detta och prova på något av våra exempel.

/stig